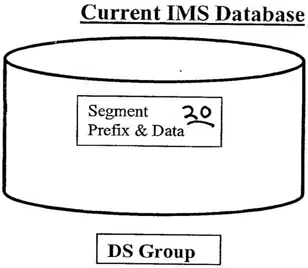

Fig 1A
(Prior Art)

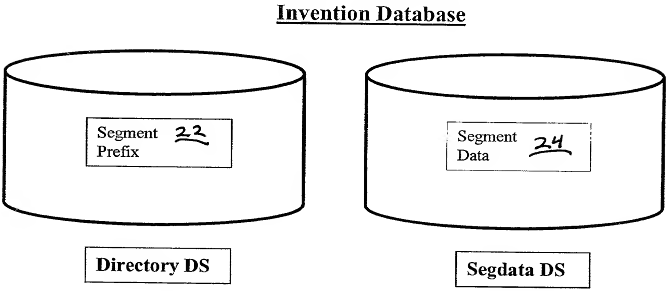

Fig 1B

Layout of Segment in Directory Dataset

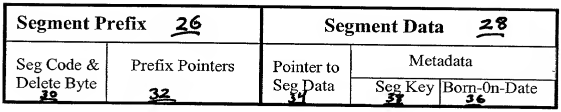

Figure 2A. Split Segment Composition – Prefix Portion with Metadata in segment data portion

Layout of Segment in Segdata Dataset

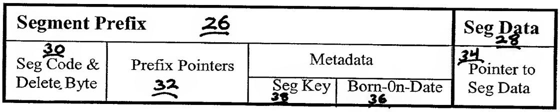

Figure 2B. Split Segment Composition – Prefix Portion with Metadata in segment prefix portion

Layout of Segment in Segdata Dataset

Segment Prefix 40	Segment Data 42	Trans-parent 44
Seg code & delete byte 46	User Data 48	Born on Date 50

Fig. 3

```
DBD     NAME=IVPDB1,ACCESS=(HIDAM,OSAM)                                         122

DIR     DD1=DFSIVD1,SIZE=2048,UOW=(500,50,10)

DATASET DD1=DFSIVD1A,DEVICE=3380,SIZE=2048
SEGM    NAME=A1111111,PARENT=0,BYTES=40,RULES=(LLV,LAST),PTR=(TB,CTR)
FIELD   NAME=(A1111111,SEQ,U),BYTES=010,START=00001,TYPE=C
FIELD   NAME=A9999999,BYTES=010,START=00011,TYPE=C
LCHILD  NAME=(A1,IVPDB1I),POINTER=INDX,RULES=LAST
LCHILD  NAME=(A1X,IVPDB1X),POINTER=INDX
XDFLD   NAME=AXXXXXXX,SEGMENT=A1111111,SRCH=(A9999999)
LCHILD  NAME=(C1X,IVPDB1Z),POINTER=INDX
XDFLD   NAME=CXXXXXXX,SEGMENT=C1111111,SRCH=(C9999999)

DATASET DD1=DFSIVD1B,DEVICE=3380,SIZE=4096
SEGM    NAME=B1111111,PARENT=A1111111,BYTES=(1000,50),                          X
              RULES=(LLV,LAST),PTR=(TB)
FIELD   NAME=(B1111111,SEQ,M),BYTES=010,START=00003,TYPE=C
FIELD   NAME=/SXB1
LCHILD  NAME=(B1X,IVPDB1Y),POINTER=INDX
XDFLD..NAME=BXXXXXXX,SEGMENT=B1111111,SRCH=(B1111111),SUBSEQ=(/SXB1)

DATASET DD1=DFSIVD1C,DEVICE=3380,SIZE=8192
SEGM    NAME=C1111111,PARENT=B1111111,COMPRTN=(DFSKMPX0,DATA,INIT),             X
              RULES=(LLV,LAST),PTR=(TB),BYTES=(8000,50)
FIELD   NAME=(C1111111,SEQ,U),BYTES=010,START=00003,TYPE=C
FIELD   NAME=C9999999,BYTES=010,START=00011,TYPE=C

DIRGEN

DBDGEN
FINISH
END
```

Figure 4A Sample HIDAM DBD

124

```
DBD     NAME=IVPDB2,ACCESS=HDAM,RMNAME=(DFSHDC40,4,1000)

DIR   DD1=DFSIVD2,UOW=(100,10)

DATASET DD1=DFSIVD2A,DEVICE=3380,SIZE=2048
SEGM    NAME=A1111111,PARENT=0,BYTES=40,RULES=(LLL,LAST),          X
              COMPRTN=(DFSKMPX0,DATA,INIT)
FIELD   NAME=(A1111111,SEQ,U),BYTES=010,START=00001,TYPE=C

DATASET DD1=DFSIVD2B,DEVICE=3380,SIZE=4096
SEGM    NAME=B1111111,PARENT=A1111111,BYTES=(1000,50),             X
              RULES=(LLV,LAST),PTR=(TB)
FIELD   NAME=(B1111111,SEQ,U),BYTES=010,START=00003,TYPE=C

DATASET DD1=DFSIVD2C,DEVICE=3380,SIZE=8192
SEGM    NAME=C1111111,PARENT=B1111111,COMPRTN=(DFSKMPX0,DATA,INIT),
        RULES=(LLV,LAST),PTR=TB,BYTES=8000
FIELD   NAME=(C1111111,SEQ,U),BYTES=010,START=00001,TYPE=C

DIRGEN

DBDGEN
FINISH
END
```

Figure 4B Sample HDAM DBD

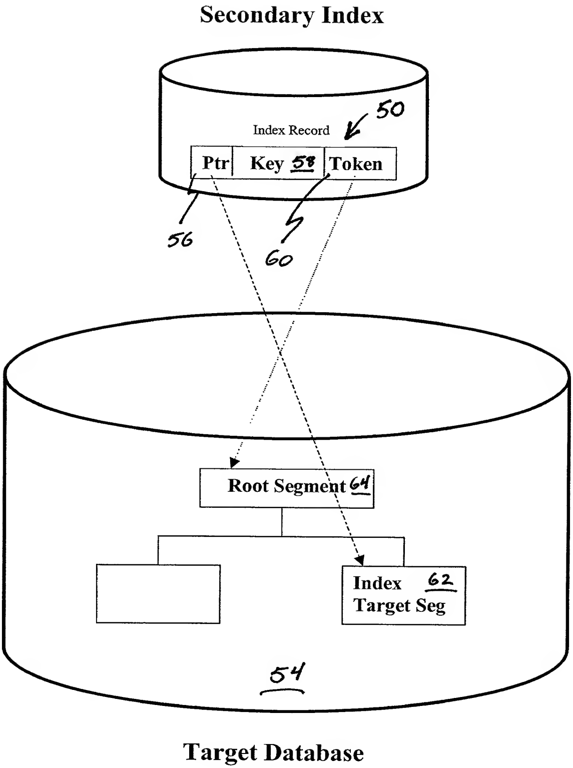

Figure 5 Secondary Index Architecture

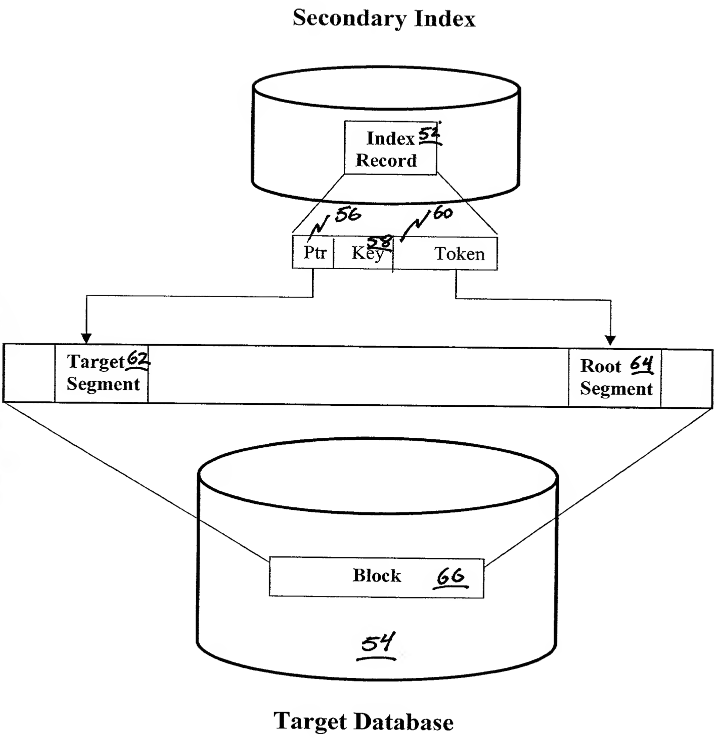

Figure 6 Secondary Index Before Reorganizing

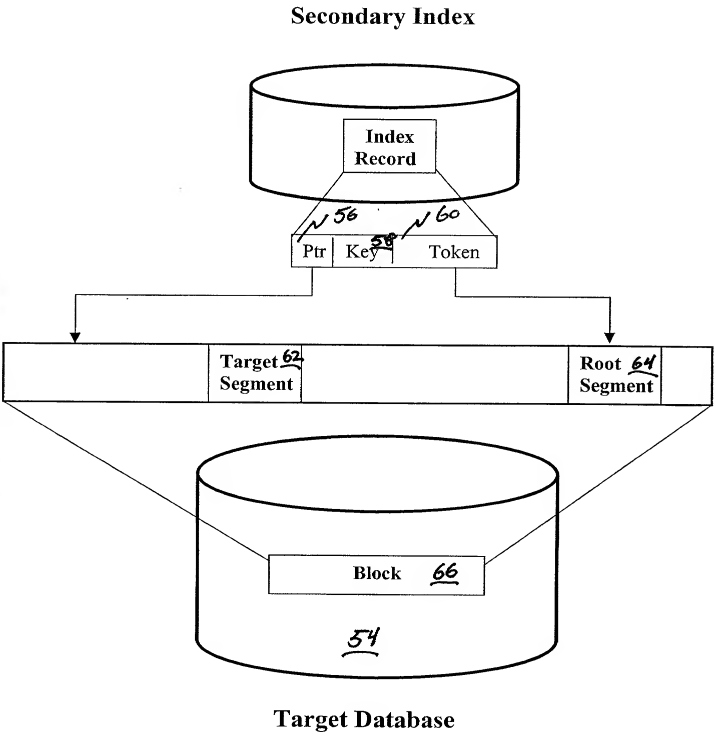

Figure 7 Secondary Index After Reorganizing

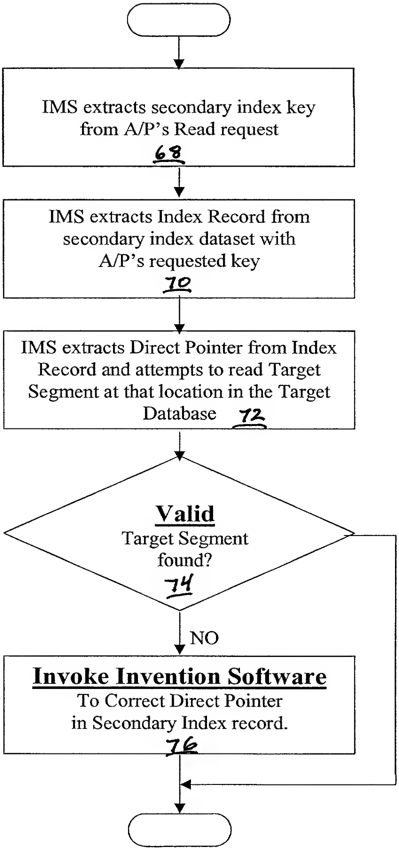
IMS extracts secondary index key from A/P's Read request
68
IMS extracts Index Record from secondary index dataset with A/P's requested key
70
IMS extracts Direct Pointer from Index Record and attempts to read Target Segment at that location in the Target Database
72
Valid
Target Segment found?
74
NO
Invoke Invention Software
To Correct Direct Pointer in Secondary Index record.
76

Figure 8 Retrieving a Target Segment via a Secondary Index

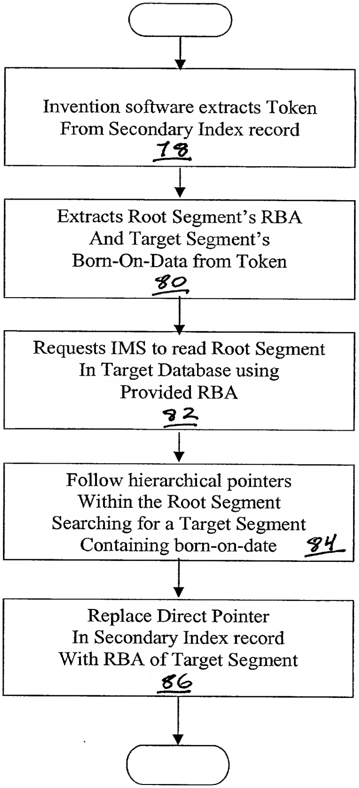
Invention software extracts Token
From Secondary Index record
78
Extracts Root Segment's RBA
And Target Segment's
Born-On-Data from Token
80
Requests IMS to read Root Segment
In Target Database using
Provided RBA
82
Follow hierarchical pointers
Within the Root Segment
Searching for a Target Segment
Containing born-on-date
84
Replace Direct Pointer
In Secondary Index record
With RBA of Target Segment
86

Figure 9 Correcting Direct Pointer in a Secondary Index

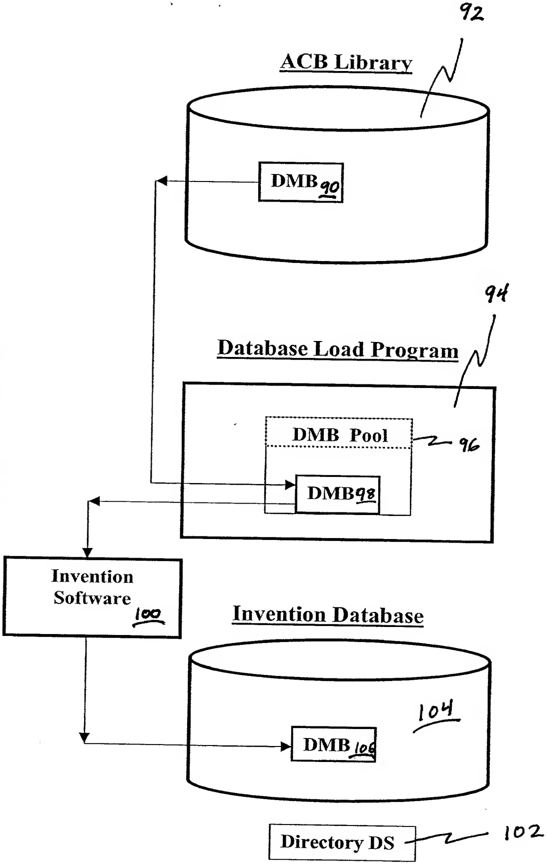

Figure 10 Saving the Database Definition at DB Load Time

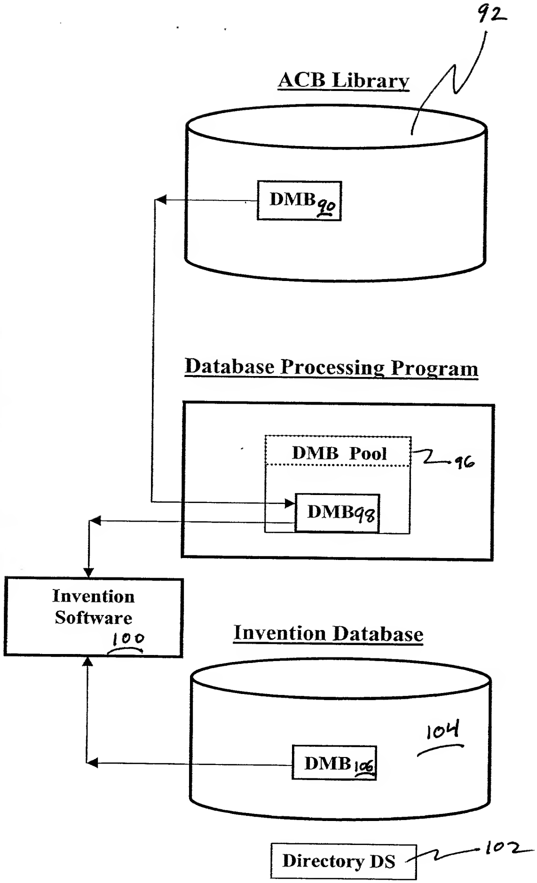

Figure 11 Checking the Database Definition at DB Processing Time

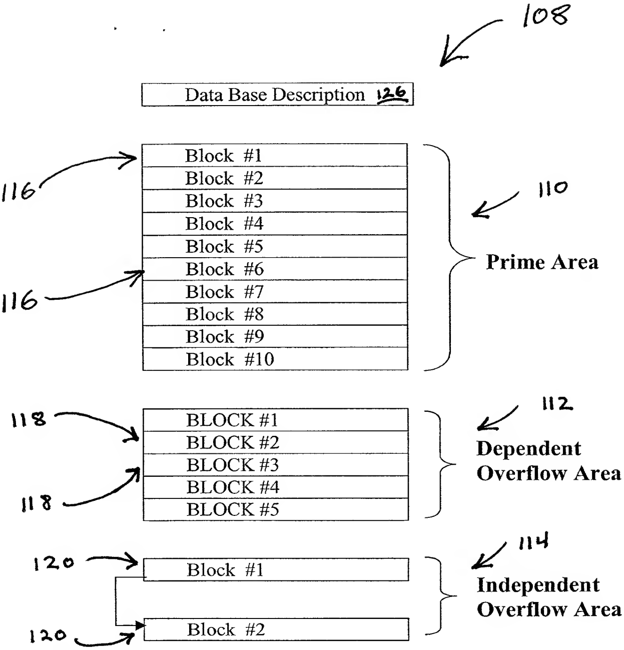

Figure 12. Unit Of Work (UOW) Architecture

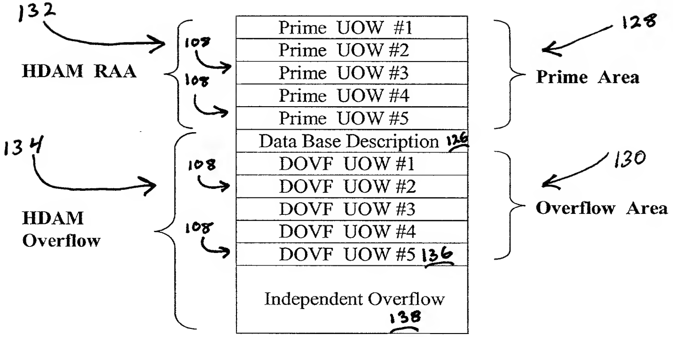

Figure 13. HDAM UOW Architecture

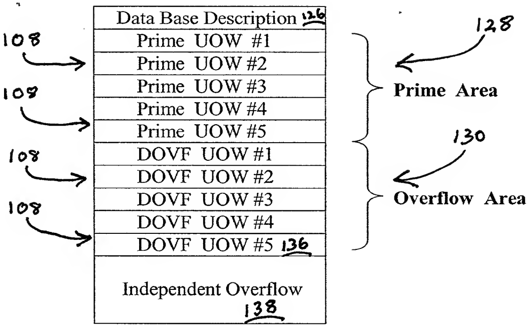

Figure 14. HIDAM UOW Architecture

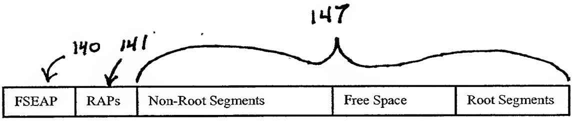

Figure 15. Prime & DOVF Block Composition

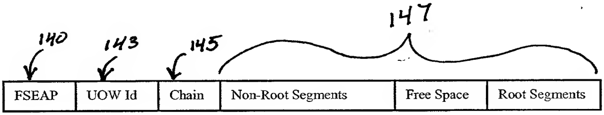

Figure 16. IOVF Block Composition

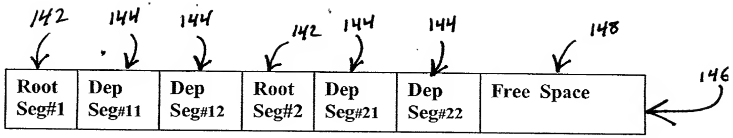

Figure 17 Block Composition Using IMS' Space Management

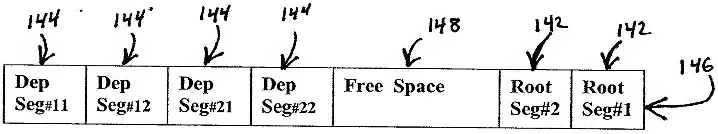

Figure 18 Block Composition Using Invention's Space Management

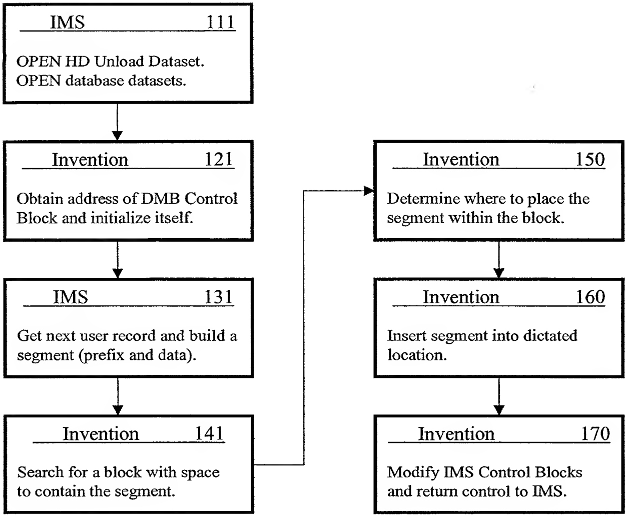
IMS 111
OPEN HD Unload Dataset.
OPEN database datasets.
Invention 121
Obtain address of DMB Control Block and initialize itself.
IMS 131
Get next user record and build a segment (prefix and data).
Invention 141
Search for a block with space to contain the segment.
Invention 150
Determine where to place the segment within the block.
Invention 160
Insert segment into dictated location.
Invention 170
Modify IMS Control Blocks and return control to IMS.

Figure 19 Space Management at Database Load Time

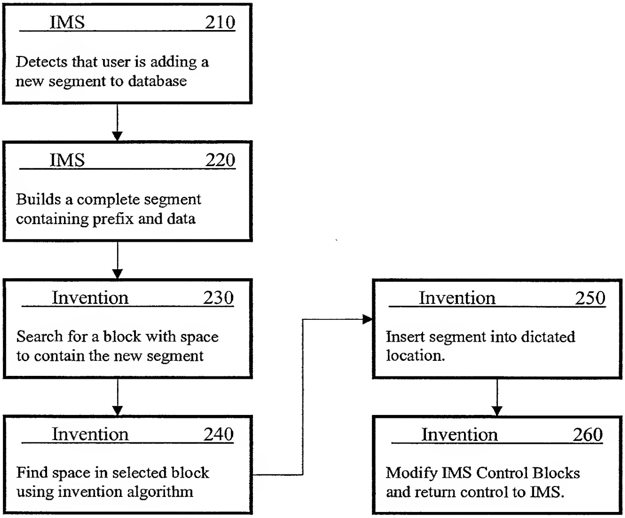
IMS 210
Detects that user is adding a new segment to database
IMS 220
Builds a complete segment containing prefix and data
Invention 230
Search for a block with space to contain the new segment
Invention 240
Find space in selected block using invention algorithm
Invention 250
Insert segment into dictated location.
Invention 260
Modify IMS Control Blocks and return control to IMS.

Figure 20 Space Management at Database Update Time

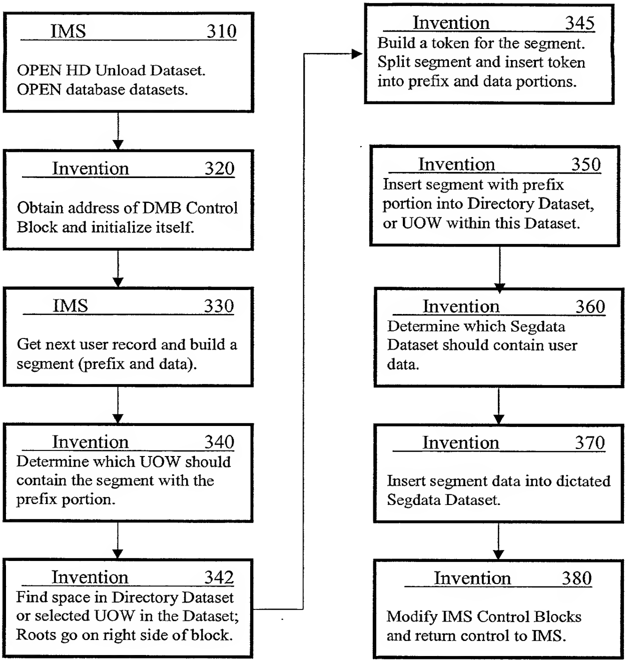
IMS 310
OPEN HD Unload Dataset.
OPEN database datasets.
Invention 320
Obtain address of DMB Control Block and initialize itself.
IMS 330
Get next user record and build a segment (prefix and data).
Invention 340
Determine which UOW should contain the segment with the prefix portion.
Invention 342
Find space in Directory Dataset or selected UOW in the Dataset; Roots go on right side of block.
Invention 345
Build a token for the segment. Split segment and insert token into prefix and data portions.
Invention 350
Insert segment with prefix portion into Directory Dataset, or UOW within this Dataset.
Invention 360
Determine which Segdata Dataset should contain user data.
Invention 370
Insert segment data into dictated Segdata Dataset.
Invention 380
Modify IMS Control Blocks and return control to IMS.

Figure 21. Space Management at Database Load Time

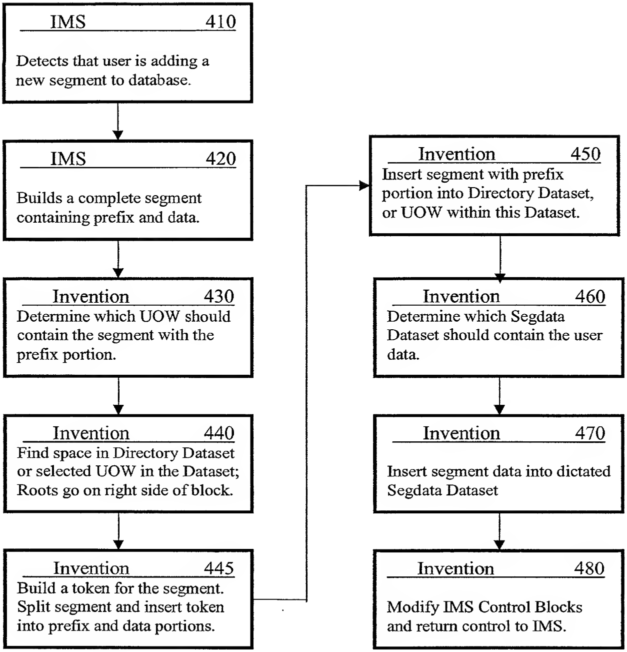
IMS 410
Detects that user is adding a new segment to database.
IMS 420
Builds a complete segment containing prefix and data.
Invention 430
Determine which UOW should contain the segment with the prefix portion.
Invention 440
Find space in Directory Dataset or selected UOW in the Dataset; Roots go on right side of block.
Invention 445
Build a token for the segment. Split segment and insert token into prefix and data portions.
Invention 450
Insert segment with prefix portion into Directory Dataset, or UOW within this Dataset.
Invention 460
Determine which Segdata Dataset should contain the user data.
Invention 470
Insert segment data into dictated Segdata Dataset
Invention 480
Modify IMS Control Blocks and return control to IMS.

Figure 22. Space Management at Database Update Time